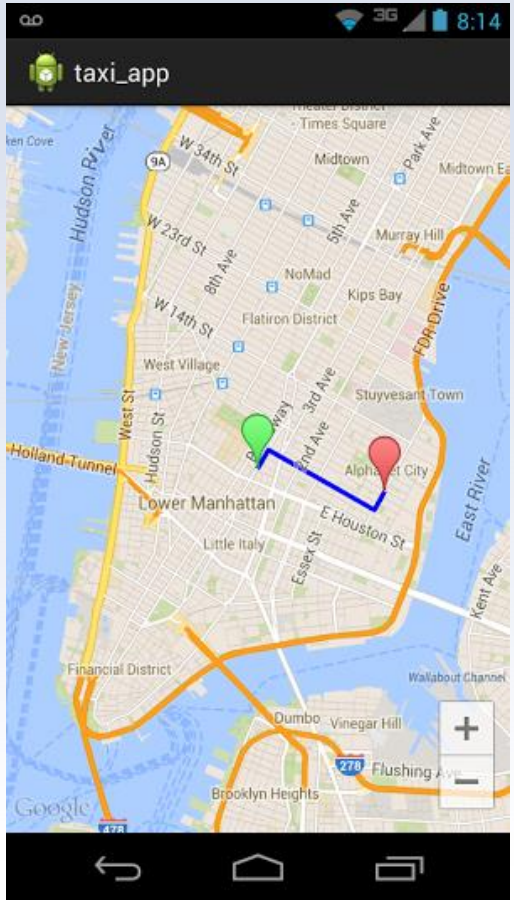
New York City Taxi Data
Data Exploration and Seek Time Prediction

Dan Work, UIUC
Brian Donovan, UIUC
Raphael Stern, UIUC
Thomas Breier, TUM/UIUC

Long Term Goal of the Project



- Give drivers recommendations where to pick up passengers as fast as possible
- →Improves overall system efficiency
- Real-time system using historic data

Overview

- Part A: Data Exploration
  - Data Set
  - Interesting Finds
- Part B: Seek Time Prediction
  - Introduction to Kernel Smoothing
  - Applied Kernel Smoothing

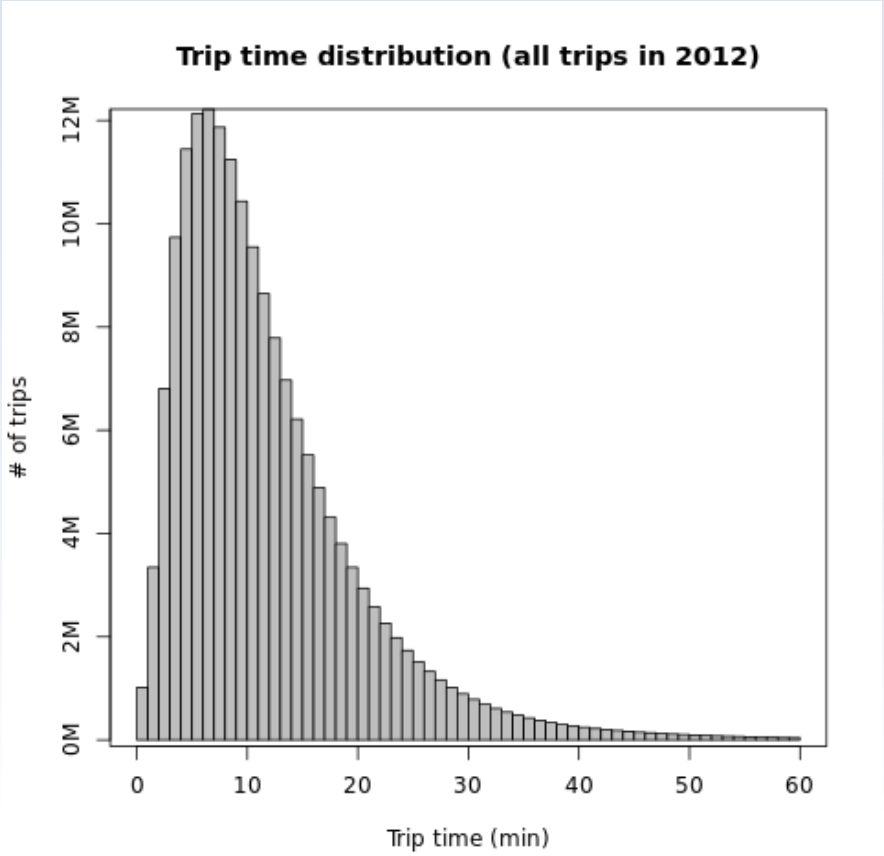
Part A

DATA EXPLORATION

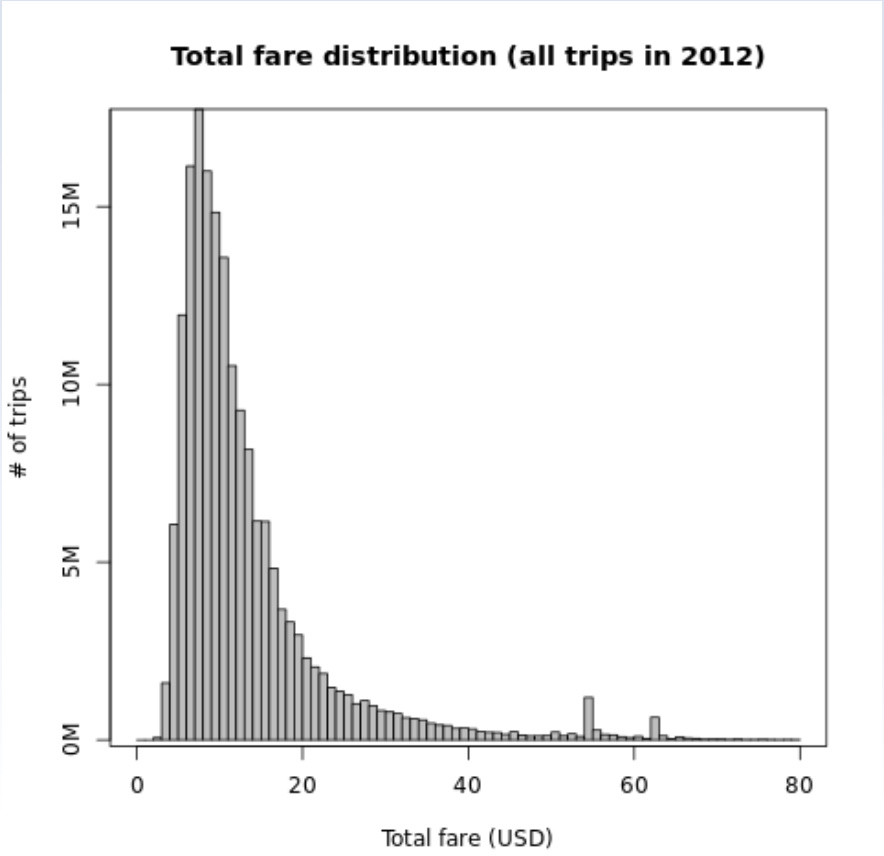
Data Set

- Retrieved by FOIA request from NYC Taxi & Limousine Comission
- All taxi trips 2010-13
- 700 Mio trips
- ~ 150 GB of .csv files
- Lots of errors and inconsistencies….
- Takes hours to days to process

Trip Time Distribution



Total Fare Amount Distribution



Seek Time

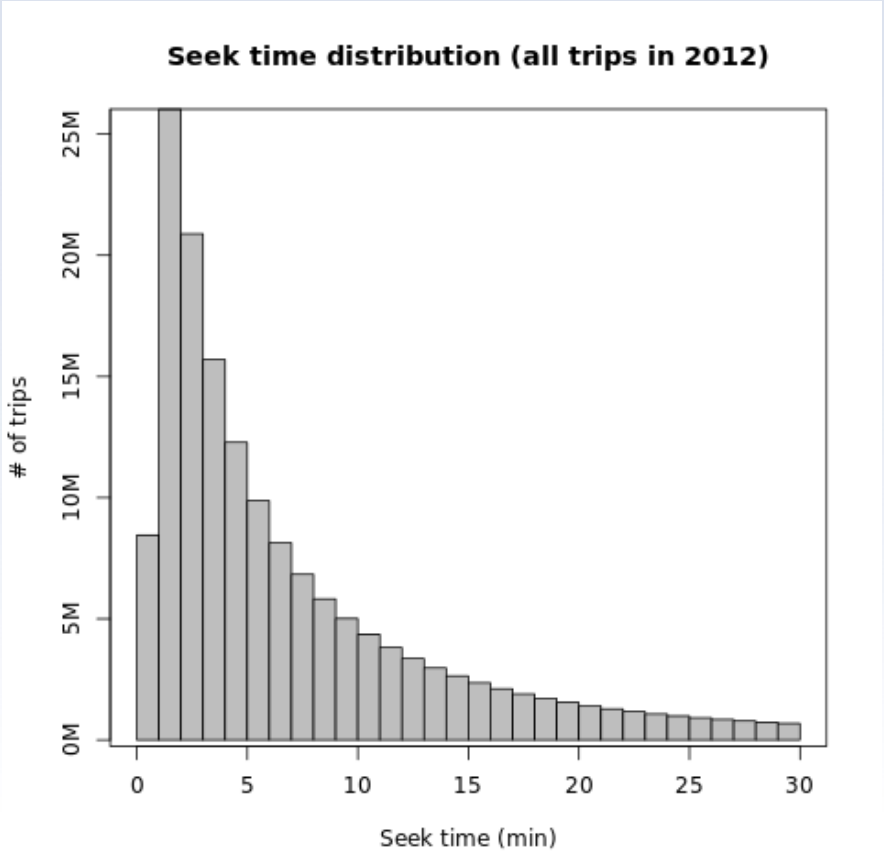
The time it takes a taxi driver to find a new passenger when working

For example:

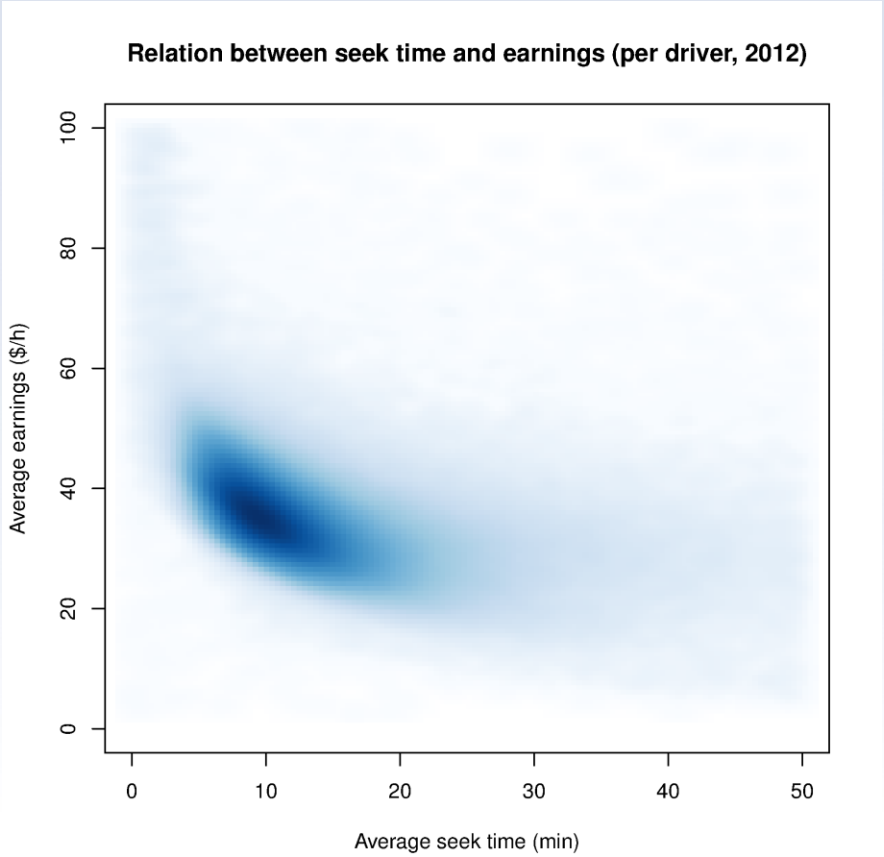
Cab driver A drops off passengers at 1:37pm and picks up his next passenger at 1:45pm

→ Seek time: 8 minutes

Seek Time Distribution



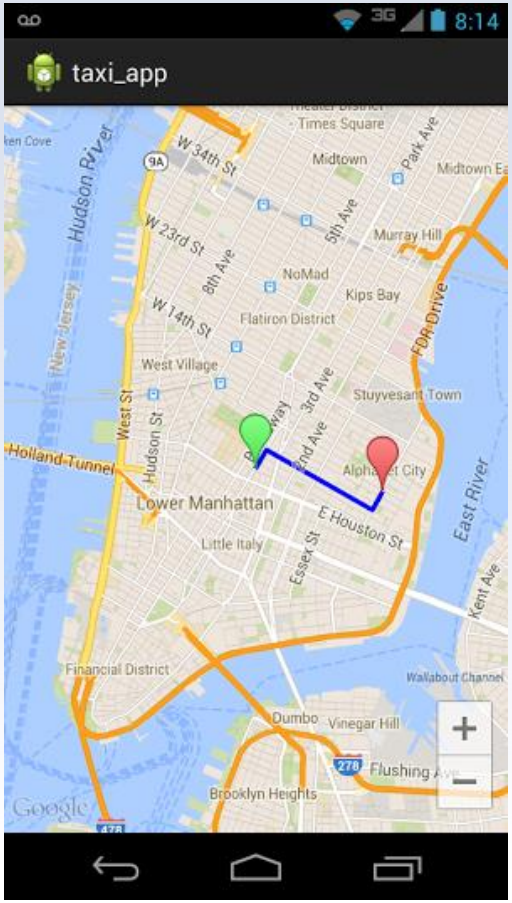
Seek Time versus Earnings



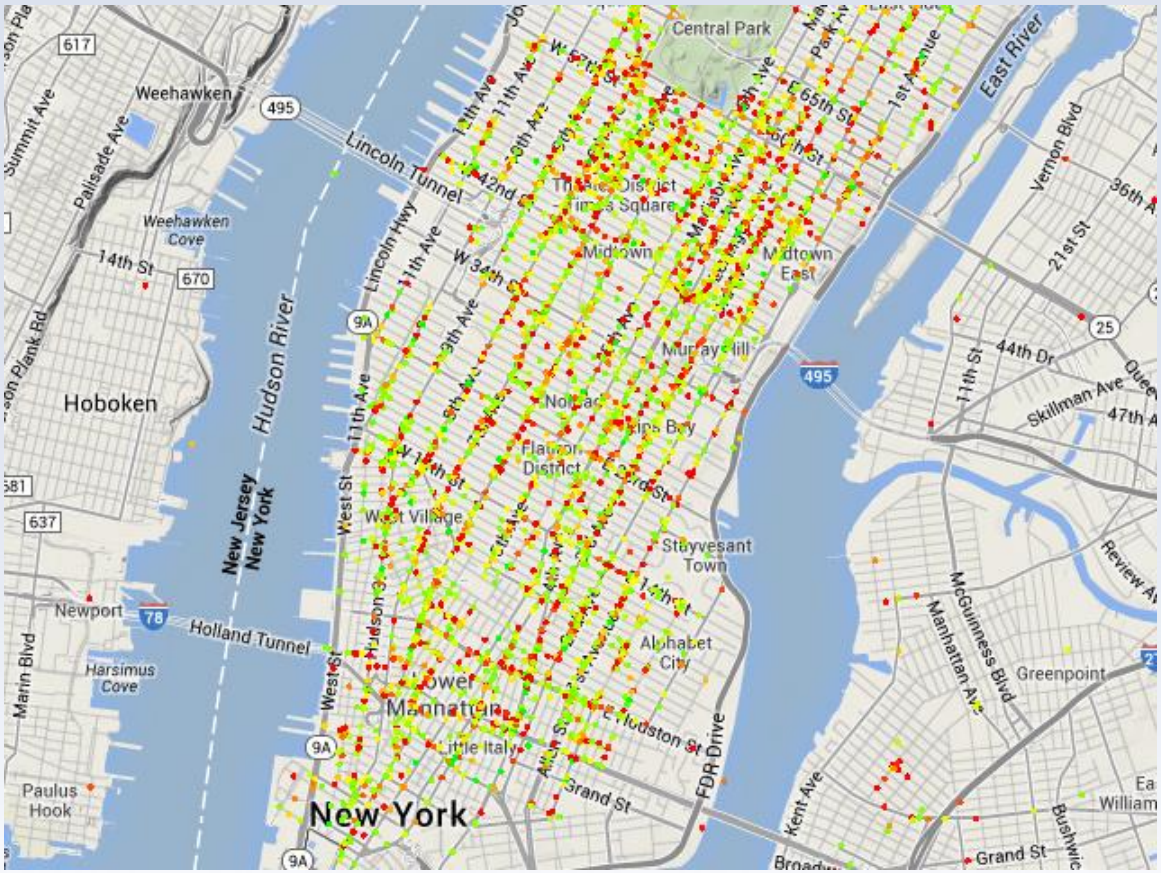
Part B

SEEK TIME PREDICTION

Seek Time Prediction – Why?



- Remember: Goal is to pick up passengers as fast as possible
- → Equal to reduce seek time
- → Try to navigate cabs in a direction to reduce seek time
- But: Can‘t measure current demand
- → Hence, predict seek time (learnt using historic data) by looking at current time slice

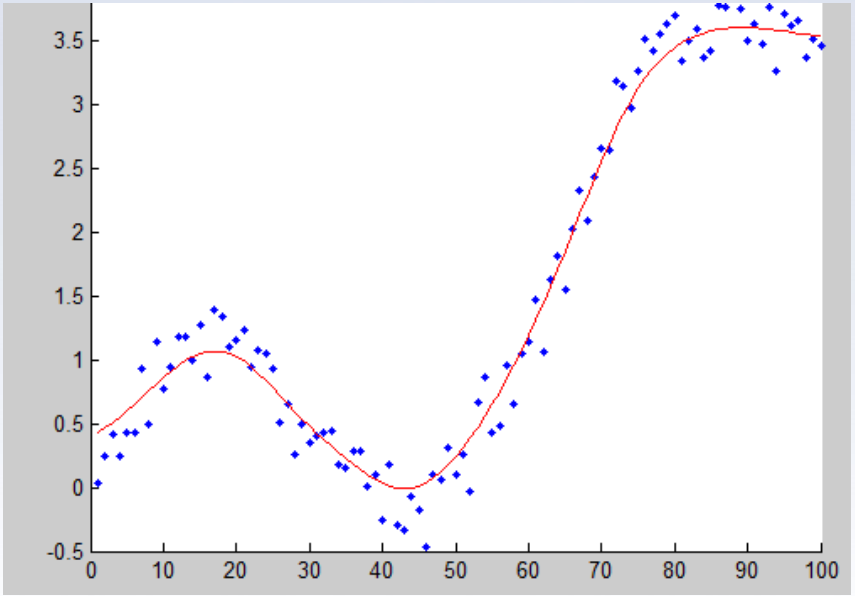


Kernel Smoothing

Will be used to determine seek time in a certain location

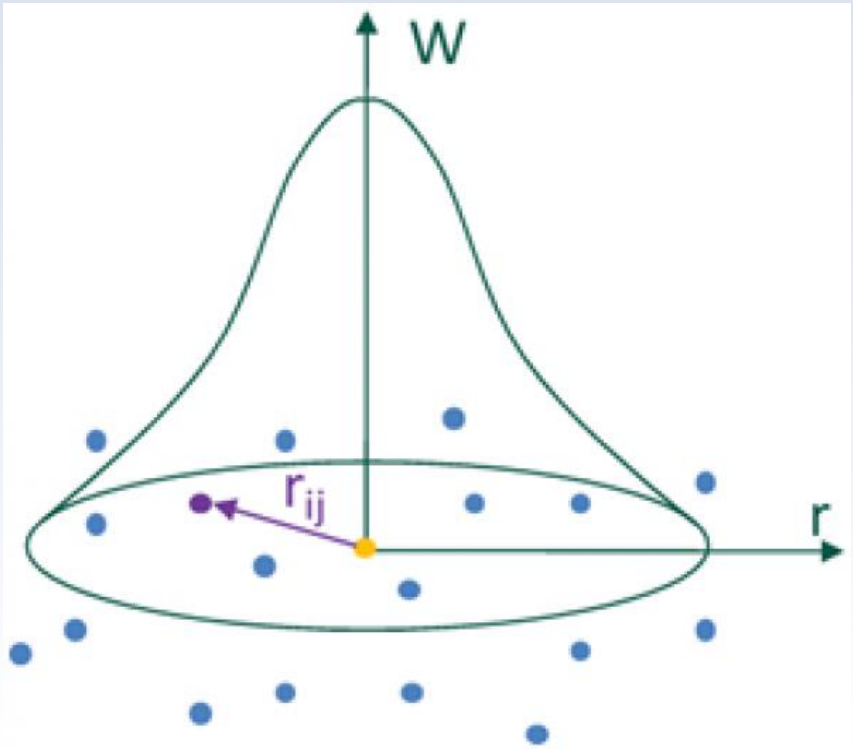
Kernel Smoothing

- Estimate a function f(x), e.g. a regression function
- Use weighted average of (all/neighboring) datapoints
- That weight is determined by a „kernel (function)“



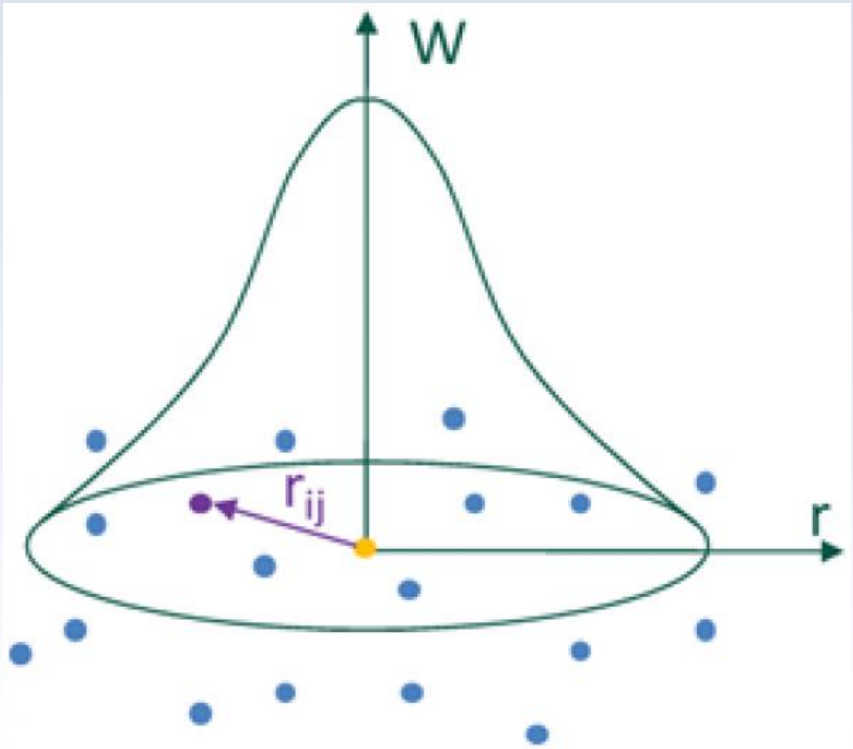
Kernel Smoothing

- Estimate a function f(x), e.g. a regression function
- Use weighted average of (all/neighboring) datapoints
- That weight is determined by a „kernel (function)“



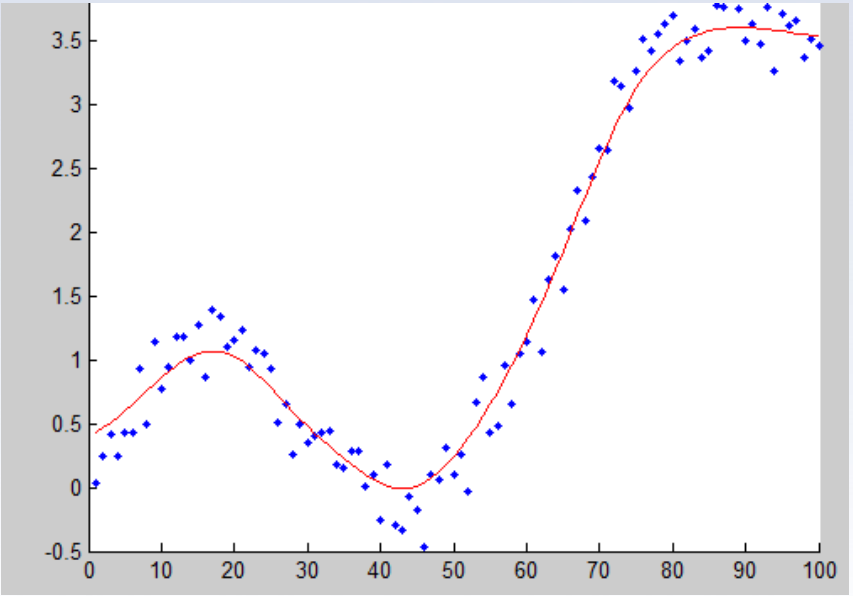
Kernel

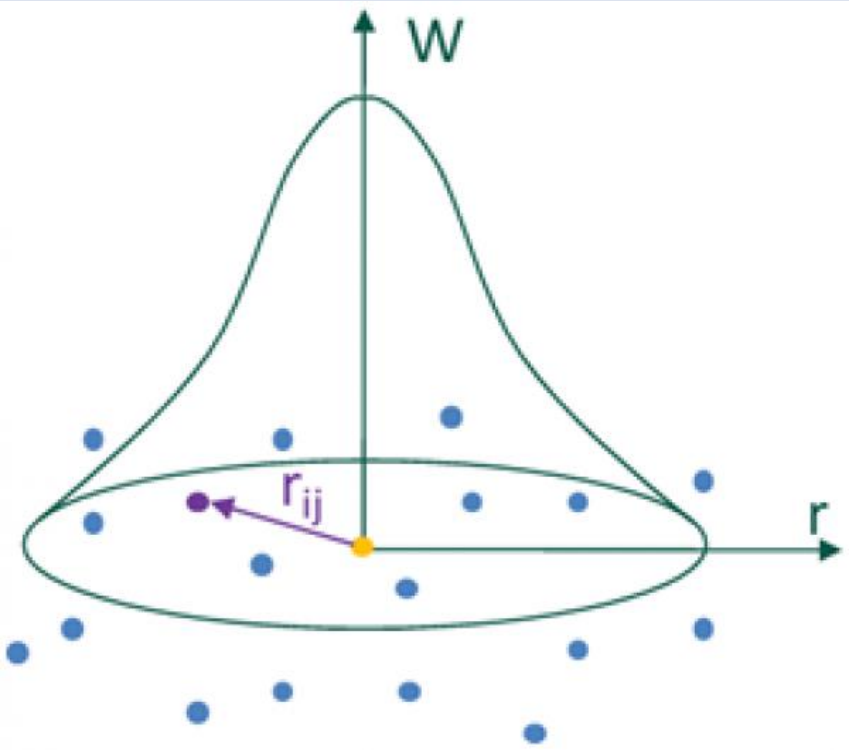
- Popular:
  Gaussian (radial basis function) kernel
- $w(r_{ij}) = e^{-(\frac{r_{ij}}{\sigma})^2}$
- Need to determine parameter σ



(Gaussian) Kernel Smoothing

- Estimate seek time S_x at position x by calculating the weighted average of the seek times of all other points
- $E[S_x] = \frac{\sum_i w(ix) \cdot S_i}{\sum_i w(ix)}$

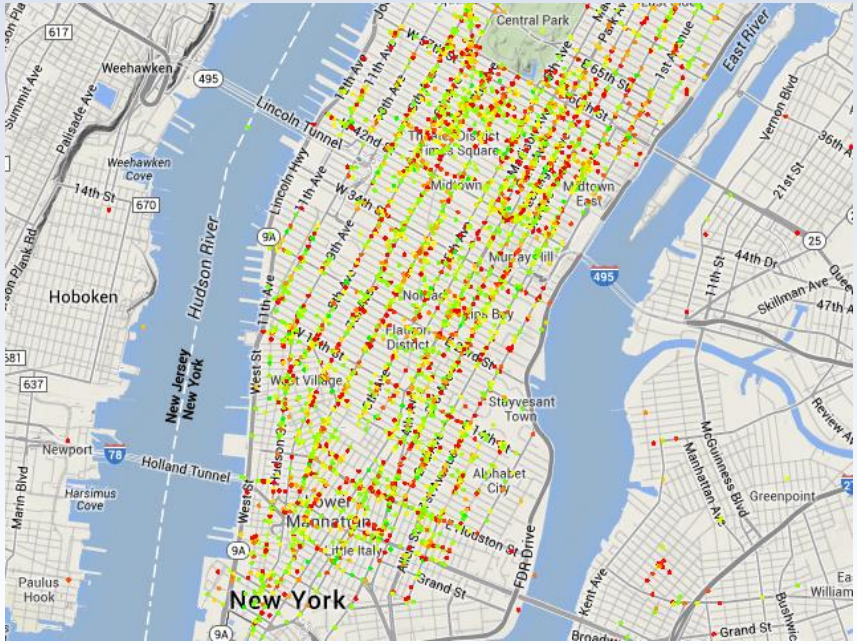




Applying Kernel Smoothing

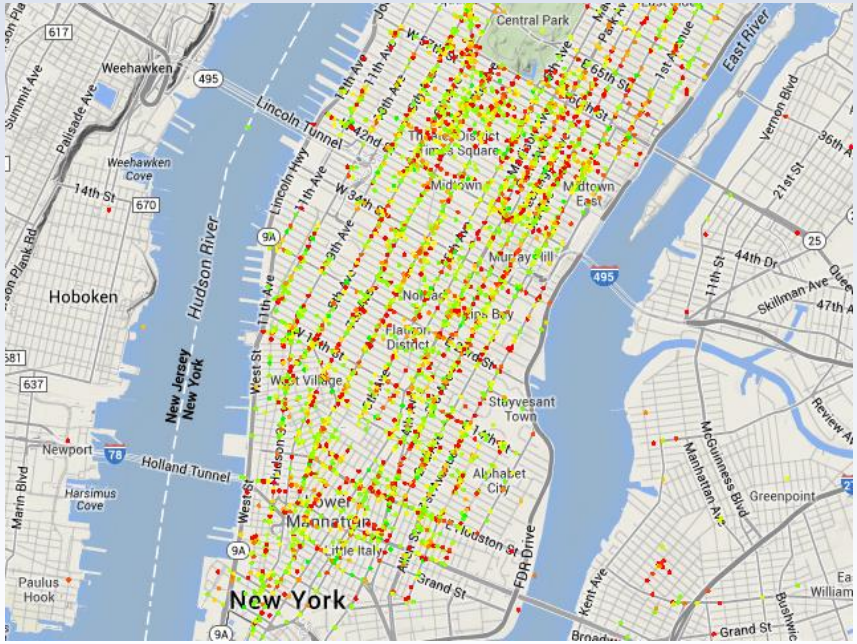
Applying Kernel Smoothing

- Evaluate Kernel Smoother for a set of fixed anchor points
- Evaluate not only for seek time, but other features as well
- Learn how seek time changes based on features → Makes prediction possible



Choosing a Kernel

- $w(r_{ij}) = e^{-(\frac{r_{ij}}{\sigma})^2}$
- Need to find a good σ
- Try $\sigma \in \{0.5, 1, \ldots, 8\}$ using leave-one-out cross-validation on several time slices
- Compare RMSE with baseline (average seektime in the whole city)

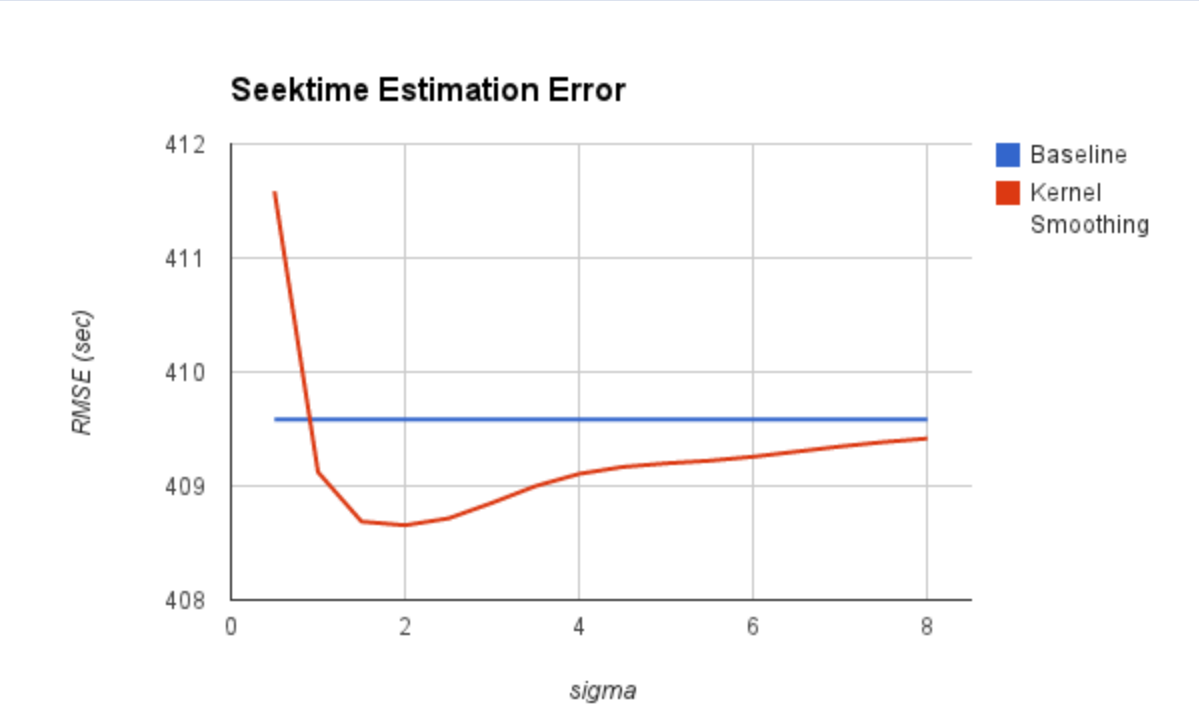


Choosing a Kernel

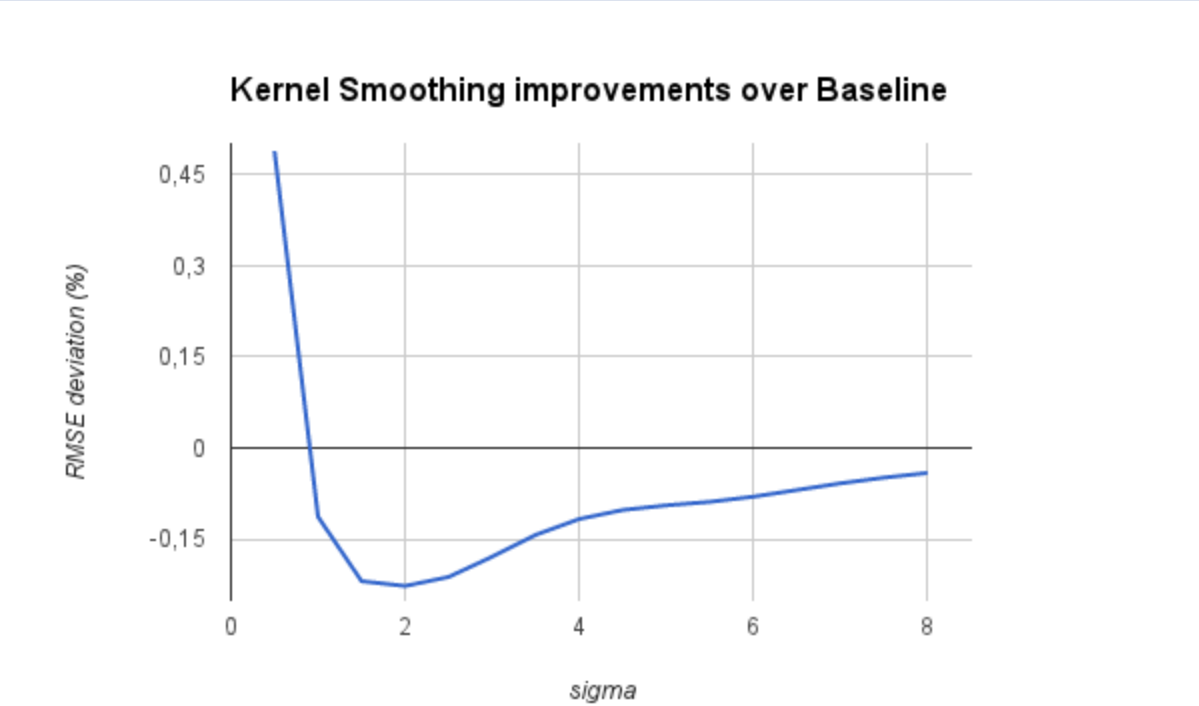
- $w(r_{ij}) = e^{-(\frac{r_{ij}}{\sigma})^2}$
- Try $\sigma \in \{0.5, 1, \ldots, 8\}$ using leave-one-out cross-validation on several time slices
- Compare RMSE with baseline (average seektime in the whole city)
- **Expectation: Kernel smoothing leads to significantly lower RMSE**

Well…

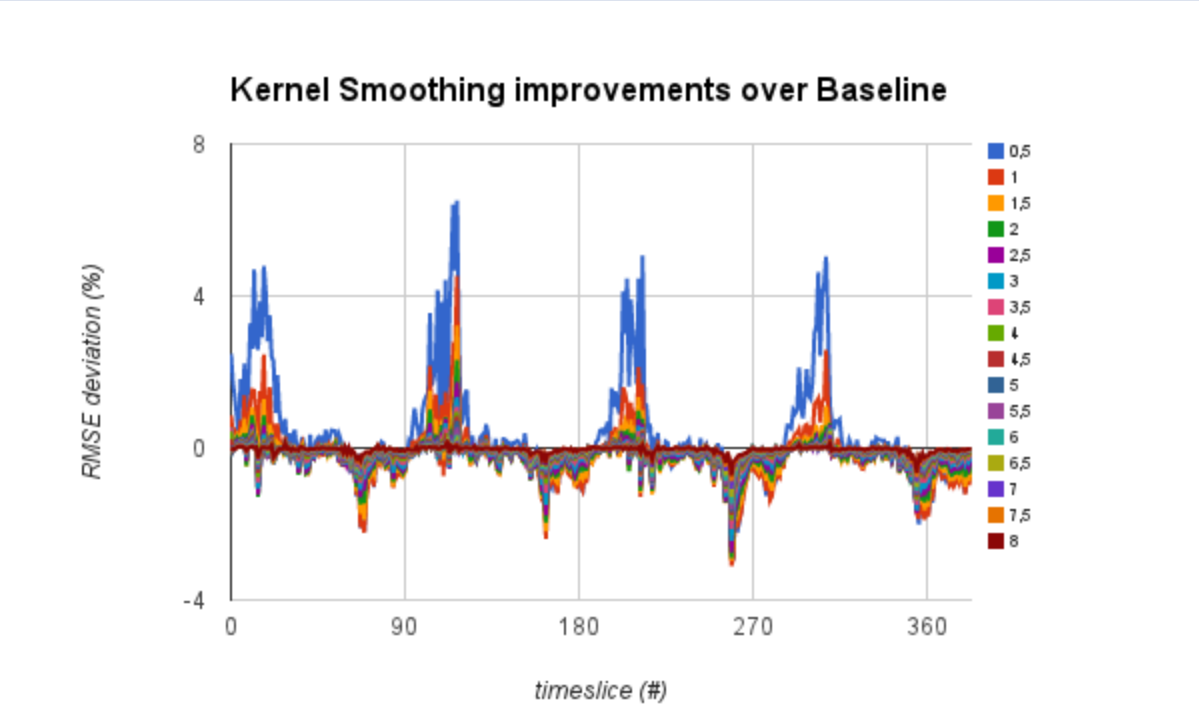
Trial 1: 15 minute time slices



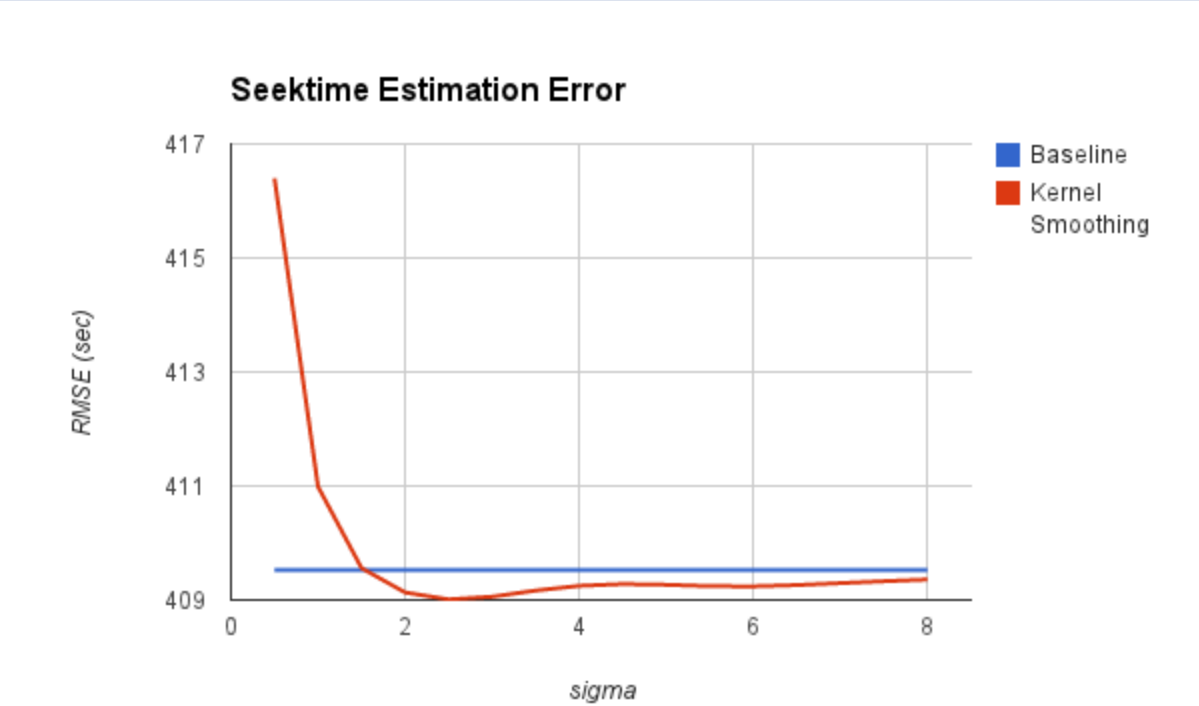
Trial 1: 15 minute time slices



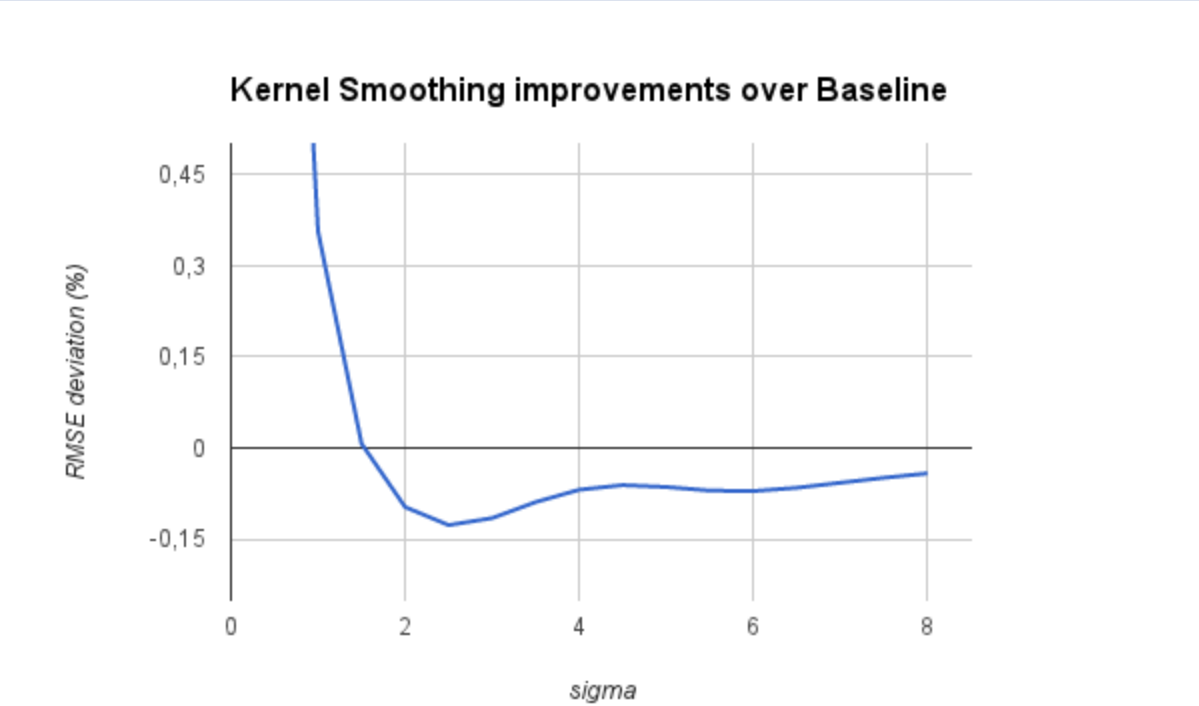
Trial 1: 15 minute time slices



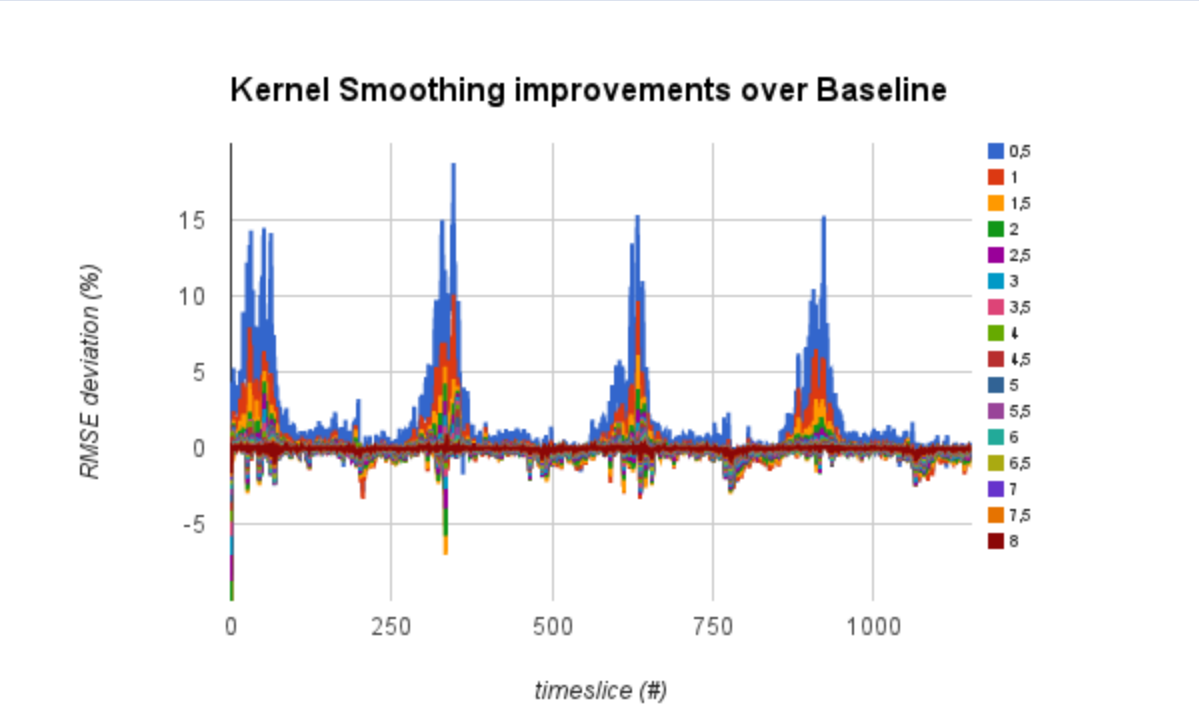
Trial 2: 5 minute time slices



Trial 2: 5 minute time slices



Trial 2: 5 minute time slices



Conclusion

- There appears to be relation between location and seek time, but it is not significant enough
- Is a different approach necessary?
- Might including other features help?

Images

- Slide 2, 12: Brian Donovan
- Slide 14, 17: http://en.wikipedia.org/wiki/File:Gaussian_kernel_regression.png
- Slide 15, 16, 18: http://www.ged.rwth-aachen.de/index.php?cat=Tools_n_Methods&subcat=Geomechanical_modeling&page=Smoothed_Particle_Hydrodynamics
- Maps provided by Google Maps
- Otherwise: Own work

Thank you!